A Turbo Decoder Stopping Criterion Improvement

Background and Summary of the Invention

10   The present application relates to error control systems in data transmission, especially to iterative decoding using parallel concatenated encoders, such as turbo decoders.

Background: Signals and Noisy Channels

15   No communications channel is perfect. In the real world, every communications channel includes some "noise" (unpredictable non-signal inputs). The amount of noise is usually quantified by the signal-to-noise ratio, or "SNR", which is stated in decibels. (For example, if the signal power is one hundred times the noise power, the SNR would be about 20 dB.) Higher SNRs are more desirable, but the SNR (at a given transmission rate) is set by the 20 physics of the channel, as well as by the design of the transmitter and receiver. Even though digital signals are generally less susceptible to degradation than analog signals, many channels include enough noise to induce some errors in a digital signal. In many applications a desired data rate can easily be achieved at reasonable cost, but the worst-case level of bit errors with such a configuration 25 is excessive. In such cases coding techniques can be used to reduce the error rate (at the cost of a slightly reduced gross data rate).

    Similar techniques can also be used with data storage architectures. In this case, the "channel" is the storage medium and the interfaces to it. For example, in a hard disk drive the write head is an analog driver stage which 30 may not be perfectly aligned to the desired radial position on the disk. Data can be degraded, during writing, storage, or reading, by factors such as mistracking, overtemperature, particulate contamination, or mechanical failure.

Background: Error-Control Coding

5

Coded digital communication systems use error control codes to improve data reliability at a given signal-to-noise ratio (SNR). For example, an extremely simple form (used in data storage applications) is to generate and transmit a parity bit with every eight bits of data; by checking parity on each

10 block of nine bits, single-bit errors can be detected. (By adding three error-correction bits to each block, single-bit errors can be detected and corrected.) In general, error control coding includes a large variety of techniques for generating extra bits to accompany a data stream, allowing errors in the data stream to be detected and possibly corrected.

15

A famous existence proof (Shannon's noisy channel coding theorem) states that error rates can be made as low as desired by using optimal error control codes. This theorem did not itself tell how to derive the optimal error control coding, and for decades no coding schemes were able to come close to the theoretical limits defined by this theorem. However, a major breakthrough

20 was achieved in 1993, when turbo codes were introduced.

Background: "Turbo" Coding

The encoder side of a turbo coding architecture typically uses two encoders, one operating on the raw data stream and one on a shuffled copy of

25 the base data stream, to generate two parity bits for each bit of the raw data stream. The encoder output thus contains three times as many bits as the incoming data stream. This "parallel concatenated encoder" (or "PCE") configuration is described in detail below.

The most surprising part of turbo coding was its decoding architecture.

30 The decoder side invokes a process which (if the channel were noiseless) would merely reverse the transformation performed on the encoder side, to reproduce the original data. However, the decoder side is configured to operate on soft estimates of the information bits and refines the estimates through an iterative

5   reestimation process. The decoder does not have to reach a decision on its first pass, but is generally allowed to iteratively improve the estimates of the information bits until convergence is achieved. (A more detailed description of this is provided below.)

One drawback to the turbo decoder is that some received codewords

10   require many iterations to converge. The ability to improve estimates by iteration is a great strength, since it means that the decoding performance can be nearly optimal; but every iteration requires additional time, computing resources, and/or battery energy. Thus in practice it is desirable to find a way to stop turbo decoding as soon as convergence is achieved. Worst-case channel

15   conditions may require many more iterations than do best-case conditions, and it is desirable to find some way to adjust the number of iterations to no more than required.

Background: Stopping Criteria for Turbo Decoders

20   Originally, turbo decoders would execute a specific number of iterations regardless of the number of errors inserted by the channel. This was inefficient, since fewer iterations are needed when there are fewer errors in the incoming signal. Efforts have therefore been made to develop adaptive stopping criteria, which would provide an indicator of when the turbo decoding

25   iterations can be stopped.

U.S. patent 5,761,248 describes adaptive stopping based on cross entropy, or convergence of the output bits: when the decoder's digital output bit estimates (for some block of data) stop changing, the decoding process for that block is halted. However, this stopping criterion requires at least two iterations

30   of the decoding process (in order to generate enough outputs to compare to one another). This can be a waste of processing effort in systems where the channel introduces few errors, or where there is a large amount of data to be decoded. Other stopping criteria, based on the cross entropy criterion, abort decoding

5   based on the ratio of sign changes in the extrinsics to the number of extrinsics.

Another attempt at adaptive stopping was described in Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," 1994 GLOBECOM Proceedings 1298, which suggests that the VARIANCE of the estimates can be used for a stopping

10   criterion.

Turbo Decoding with Improved Adaptive Stopping

The present application teaches an improvement to a stopping criterion presented in commonly-owned U.S. application number _____, which has

15   the same effective filing date as the present application, and which also has overlapping inventorship with the present application, and which is hereby incorporated by reference for all purposes.

That application teaches an innovative method of aborting the decoding iteration process. Instead of a comparison between successive outputs of the

20   MAP decoders, an absolute indication of the decoded signal quality is compared with a threshold signal quality. The decoders generate soft outputs, or estimates of the original transmitted data symbols. With each decode, these extrinsic values diverge (or their absolute value increases), which indicates greater certainty of the value of the transmitted bit. The values of the extrinsics

25   are used to compute an estimate of the overall signal quality. The mean and variance of the extrinsic distribution are used to compute the estimate. When the signal quality reaches the desired performance threshold, iterations are aborted.

Implementation of the comparison of the decoded signal quality to the

30   threshold requires 3 divisions, 1 multiplication, 1 subtraction, and 1 comparison. All these functions are easily built in a hardware solution except the division function. Typically, division functions A/B are implemented as A x 1/B, with 1/B implemented as a lookup table. This solution only works if the

5   range of numbers for B is relatively small. Unfortunately, the previous solution for comparing the SNR required division by N (the number of extrinsics), which can be very large, often ranging from 320 to 5120. This division also has numerators that can range from 10^2 to 10^6. The range for the required divisions is therefore many orders of magnitude.

10   The present application teaches a stopping criterion implementation that does not require division by a variable. By manipulating the comparison equation, the previously required division functions are replaced by multiplications. This innovation greatly simplifies hardware implementation and improves processing speed.

Brief Description of the Drawings

The disclosed inventions will be described with reference to the accompanying drawings, which show important sample embodiments of the invention and which are incorporated in the specification hereof by reference, wherein:

Figure 1 shows a block diagram for parallel encoders.

Figure 2 shows a block diagram for a recursive systematic encoder.

Figure 3 shows a block diagram of a turbo decoder.

Figure 4 shows a stopping criterion implementation using division by a variable quantity N.

Figures 5a and 5b show graphs of SNR calculated from soft-decision output values as MAP decodes are executed.

Figure 6 shows a stopping criterion implementation without division by N.

5 Detailed Description of the Preferred Embodiments

The numerous innovative teachings of the present application will be described with particular reference to the presently preferred embodiment. However, it should be understood that this class of embodiments provides only a few examples of the many advantageous uses of the innovative teachings
10 herein. In general, statements made in the specification of the present application do not necessarily delimit any of the various claimed inventions. Moreover, some statements may apply to some inventive features but not to others.

15 Additional Background

Coded digital communication systems use error control codes to improve data reliability. Both the physical propagation medium (the channel) and the hardware used to receive a signal introduce error in the form of noise or signal degradation.
20 Decoding makes use of redundant bits added to a data stream to detect errors and estimate the most probable value for data bits given the other bits received. The effectiveness of error control codes is expressed in terms of coding gain, which is the difference between the E_b/N_o needed to achieve a given bit error rate (BER) in a coded system and the E_b/N_o required to achieve
25 the same BER in an uncoded system.

Though Shannon's noisy channel coding theorem states that arbitrarily low BERs are possible using error control codes, no coding schemes were able to achieve gain near the Shannon limit until 1993 when turbo codes were introduced. Turbo coding provides error control performance to within a few
30 tenths of a decibel of the Shannon limit.

The use of parallel concatenated encoders and iterative decoders distinguishes turbo codes from other coding systems. A PCE consists of two or more recursive systematic encoders (either block or convolutional encoders).

5   Encoders add parity bits to the original message in order to detect the presence of errors in the message. (In a rate 1/2 convolutional encoder, for instance, two data bits are output for every input data bit.) In the case of two convolutional encoders (a typical case for turbo code schemes) an original message block is encoded using the first encoder, generating a first codeword.

10   The original message is then interleaved (i.e., the input data symbols are reproduced in a different temporal order) and the result is encoded by the second encoder, producing a second codeword. The original message and both codewords are then multiplexed and transmitted to the receiver. The component encoders are typically encoders for block codes or convolutional

15   codes. The symbols are permuted either on a block basis (a block interleaver) or on a sliding window basis (a convolutional interleaver).

Turbo decoders work by passing soft decoding information between two decoding algorithms. Each decoder concentrates on one side of the encoding process, and through successive computations the two decoders arrive at a

20   consensus on the detection of the transmitted data. One decoder executes and generates N new extrinsics (*a posteriori* probabilities for the value of a given information symbol, where N is the number of data symbols), which are then applied to the other decoder. This second decoder executes and generates N new extrinsics, which are passed back to the first decoder. This completes one

25   iteration of the decoding process.

Figure 1 shows a typical encoder setup. The original data stream, x, is encoded by two recursive systematic convolutional encoders **100, 102.** (An encoder is systematic if it sends the uncoded data as part of the coded message, as shown in the rate-1/2 recursive systematic encoder diagram of **Figure 2.**)

30   In the encoding process, x is encoded by the top encoder **100**, and x and p_1 (the first codeword) are output. The data x is also interleaved by an interleaver **104** and input to the bottom encoder **102**, which generates outputs x* and the second codeword, p_2. The original data x, p_1, and p_2 are combined

5   by combiner circuitry **106** and sent through the channel to the receiver. (Note that the interleaved x* is not sent, since it is redundant information.)

Figure 2 shows a recursive systematic convolutional encoder **200**. The binary data stream x is fed into a shift register circuit **202** consisting of a series of memory elements. With each successive input, the memory elements are

10   added according to a fixed pattern, creating a pair of outputs. One output is the unchanged data x, the other is the coded data p. These outputs can be multiplexed to create a single coded data stream.

Decoders for PCEs would be expensive and computationally inefficient without iterative decoding. Iterative decoding, or turbo decoding, is a relatively

15   low complexity decoding method that exploits the component substructure of the PCE. An iterative decoder has a soft-input/soft-output (SISO) component decoder for each component encoder of the PCE. The decoders generate separate *a posteriori* probabilities for a given information symbol. The decoders take turns decoding the received data, operating on each other's

20   incompletely decoded output. With each iteration, the estimation of the value of an information symbol improves in accuracy.

Figure 3 shows a block diagram of a turbo decoder **300**. The original message block x and code words p_1 and p_2 are scaled by $2/\sigma^2$ (a measure of the channel noise variance) and stored in three separate memories of size N as

25   $\Lambda(x)$, $\Lambda(p_1)$, and $\Lambda(p_2)$ respectively. Interleavers **302, 304** for the data are also of size N. The extrinsic memories have outputs W_1 and W_2 (the *a posteriori* probability data, or estimates of the probability distribution of a code symbol that are updated with each iteration) which are set to the initial conditions before the first decoding iteration. The scaled inputs $\Lambda(x)$ and $\Lambda(p_1)$ and the

30   deinterleaved W_2 are applied to the upper MAP decoder **308**, as shown in the diagram. This MAP decoder **308** executes and generates N new extrinsics W_1 which are stored in the extrinsic memory within the interleaver I **304**.

Interleaved $\Lambda(x)$, $\Lambda(p_2)$, and interleaved W_1 are applied to the lower MAP

5   decoder **310**. The lower MAP decoder **310** executes and generates N new extrinsics W_2 which are stored in the extrinsic memory of the deinterleaver I^{-1} **306**. This completes the first decoding iteration. At each iteration, the upper MAP decoder **308** operates on $\Lambda(x)$, $\Lambda(p_1)$, and the deinterleaved W_2 (which is updated at each iteration). The lower MAP decoder **310** operates on

10 interleaved $\Lambda(x)$, $\Lambda(p_2)$, and the interleaved W_1 (which is also updated at each iteration).

Iterations are executed until a specified number have been completed or until the iterations are aborted. When decoding is complete, $\Lambda(x)$, W_1, and deinterleaved W_2 are summed. The result is sliced, meaning the same

15 component of each vector $\Lambda(x)$, W_1, and deinterleaved W_2 are used to determine that component of the output vector x, which is the decoded message. These decoded bits are then output.

The number of iterations performed varies, typically ranging from three to twenty. Depending on the noise pattern of a given received codeword, the

20 decoder might reach its optimum solution after only one or two iterations. The remaining iterations would not improve performance and are a waste of processing power and time. If the noise pattern of a received codeword is severe, the decoder may require many iterations to reach an optimum solution.

25

Soft-Decisions

In digital communication systems, data is sent in the form of symbols (usually binary) that take on one value of a finite set of possible values. These symbols are modulated onto a carrier signal that is transmitted through the

30 channel to a receiver. In the receiver, detection circuitry examines the received signal and determines what symbols were most likely transmitted. For example, in a binary hard-decision system, the receiver determines if a zero or a one was sent, and outputs a zero or one accordingly. No information as to

5   the quality of the symbol determination is included in a hard-decision output.

Soft-decision decoding takes advantage of additional information generated by the receiver decision circuitry. Rather than assign a zero or a one to each received binary signal, soft-decision receivers use an expanded set of symbols to describe the value of a given transmitted piece of data. For

10   example, in one implementation, each binary data symbol transmitted in the original data stream is expressed as an eight-bit value. Different regions are established that indicate the strength of the decision, with intermediate values given to decisions that are less reliable.

In the preferred embodiment, the output of the encoder is binary (zeros

15   and ones). A modulator maps the ones to a positive one value and maps the zeros to a negative one value. When the receiver determines a particular received symbol, it expresses it as an eight-bit value, the "soft" decision. The closer a soft-decision gets to its extreme value, the more reliable the estimation of the original data bit. These estimates are made after each MAP decode,

20   which is a substep of a full iteration. In the preferred embodiment, one iteration consists of two MAP decodes. (Note that the number of MAP decoders required depends on the number of encoders used.)

As a turbo decoder goes through its iterations, the certainty of the value

25   of a given information symbol generally increases. Thus, with each iteration substep (i.e., each MAP decode), the value of a "zero" in the original data stream will become more and more negative (since the zeros are modulated into negative numbers) and the value of a "one" in the original data stream will become more and more positive (because the ones were modulated as positive

30   numbers).

5   **Stopping Criteria for Turbo Decoders**

A stopping criterion has been developed for turbo decoding that measures the SNR of the extrinsics and compares the result to a threshold. If the threshold is reached, no further iterations are executed. This stopping criterion is disclosed in the commonly owned U.S. application # _____,

10   referenced above.

All the functions of the stopping criterion can be easily implemented in hardware except for division, which cannot be implemented in an efficient manner. Typically, division functions like A/B are implemented as A x 1/B. 1/B is implemented as a lookup table. This solution works satisfactorily as long

15   as the range of numbers for B is small. In a previous method of turbo decoding, the stopping criterion required division by a variable quantity N. N (the number of extrinsics generated by the component decoders) can be large, ranging from 320 to 5120. Often, division by N must be performed from 10^2 to 10^6 times. It is therefore inefficient to implement this one division.

20

Turbo Decoder Stopping Criterion Improvement

The present application discloses an innovative improvement to the turbo decoding stopping criterion. The improved stopping criterion requires no division functions. By manipulating the algorithm, division by message dependent quantity N is eliminated and replaced with multiplication by a

25   constant. This greatly simplifies the execution of the algorithm and makes hardware implementation much more efficient.

Figure 4 shows a block diagram of a stopping criterion implementation. The block diagram is broken into two regions. The top half generates the two

30   moments M1 and M2, which are needed to compute the SNR for the extrinsics. M1 is the sum of the N extrinsics at any given point in the iterative decoding process, and M2 is the sum of the squares of the N extrinsics at a given point.

5   Their equations are

$$M_1 = \Sigma |W_i|$$

and

$$M_2 = \Sigma |W_i|^2$$

10   where i is summed from 1 to N. The top section is executed once for each of the N extrinsics, generating a single M_1 and a single M_2. Each component of the extrinsic W_i is input and added to the sum, which is stored in the register.

M_1 and M_2 serve as input for the bottom half of the stopping criterion. These quantities are used to compute a signal-to-noise ratio estimate from the

15   extrinsics. The extrinsics W_i are assumed to be distributed as a bimodal Gaussian probability distribution function, with one mean at +m and another mean at -m. Each mode has the same conditional variance v^2. The absolute value of the probability distribution function is taken, making a single mean at +m. An increase in this mean indicates greater certainty in the value of the

20   decoded bits. An increase in the variance would indicate less certainty in the decoded values.

Alternatively, the divergence of the decoded bits can be used. If the absolute value is not taken of the probability distribution function, the two means (the positive and negative modes) will diverge as decoding progresses.

25   An estimate of signal quality based on the mean and variance of the extrinsics is computed by dividing the m^2 (the signal) by v^2 (the noise). The mean m is estimated by

$$m = \frac{\Sigma |W_i|}{N}$$

Page 13

5   and the optimal estimate for the variance is

$$v^2 = \frac{\Sigma \, |W_i|^2}{N} - (\frac{\Sigma \, |W_i|}{N})^2$$

The bottom half of the block diagram in **Figure 4** generates the signal quality estimate, which is then compared with the threshold K. If the quality
10 estimate is greater than the threshold, then the *Exceed* signal is active. The *Exceed* signal will inform the turbo decoder to stop executing its remaining iterations because an optimum solution has been reached. The bottom half executes only once per MAP decode.

Figures 5a and **5b** show a graphical representation of the change in
15 extrinsic values as MAP decodes are executed. The "x" symbols represent SNR values calculated for the extrinsics after each MAP decode. In **Figure 5a**, 5 MAP decodes are executed before the SNR of the extrinsics exceeds the threshold. After the 5th decode, no more iterations are executed. Note that this represents only 2.5 full turbo decode iterations, since two MAP decodes
20 complete one iteration in the preferred embodiment.

Figure 5b shows an example decode for a system with fewer errors than shown in **Figure 5a**. When fewer errors are inserted in the data stream, fewer MAP decodes are necessary to reach an acceptable SNR. Thus, as shown in **Figure 5b**, only two MAP decodes produces a SNR above the threshold.
25 Decoding would stop at this point.

The following algebraic expression for the bottom half of the diagram shows the comparison of the SNR (computed from the extrinsics) with K, the threshold SNR.

$$\frac{(\frac{\sum |W_i|}{N})^2}{\frac{\sum |W_i|^2}{N} - (\frac{\sum |W_i|}{N})^2} > K$$

5

The top half of the block diagram requires 2N additions, N multiplications, and N absolute value functions. The bottom half requires 3 divisions, 1 multiplication, 1 subtraction, and 1 comparison.

Figure 6 shows the innovative stopping criterion in a block diagram.

10 The comparison equation for the bottom half of the algorithm is rewritten by multiplying the denominator and numerator by N^2. The equation can be rewritten as follows.

$$\frac{(\sum |W_i|)^2}{N \sum |W_i|^2 - (\sum |W_i|)^2} > K$$

$$(1 + \frac{1}{K}) x (\sum |W_i|)^2 > N \sum |W_i|^2$$

15

This equation is the basis for the block diagram shown in **Figure 6**. The bottom half division functions are replaced by two multiplications. One multiplier can be used and multiplexed three times for an efficient implementation. The K threshold has now been replaced with a 1 + 1/K

20 threshold. Since K is a constant, 1 + 1/K is also a constant which can be precomputed by the user or generated from a simple lookup table prior to the start of a turbo decode.

The top half of the algorithm is the same as the stopping criterion in

5   **Figure 4.** The bottom half of the diagram shows that N is no longer a denominator, which greatly simplifies hardware implementation. M₁ is squared and multiplied by 1 + 1/K. This value is compared to the product of M₂ and N.

The value of the threshold is still a constant, and when the computed
10   quality estimate of the decoded signal exceeds the threshold, the decoding iterations are stopped. This allows the decoding to continue when there are large numbers of errors introduced into the signal by the channel while it prevents the decoder from executing useless iterations when few errors are introduced.

15

Definitions:

Following are short definitions of the usual meanings of some of the technical terms which are used in the present application. (However, those of ordinary skill will recognize whether the context requires a different meaning.)
20   Additional definitions can be found in the standard technical dictionaries and journals.

Eb/No: Ratio of energy per modulating bit to the noise spectral density.

BER: Bit error rate.
25   **PCE:** Parallel Concatenated Encoder.

SI/SO: Soft-input soft-output. "Soft" input or output refers to using greater bit size (for instance, 8-bits) to describe a given data symbol.

MAP decoder: Maximum A-Posteriori. MAP decoders use a detection criterion that leads to the selection of x that maximizes the probability
30    $p(x/r)$ of a symbol x given some received information r.

Extrinsic: Outputs of decoders that estimate the value of a decoded bit. Extrinsics are usually soft estimates.

5 Modifications and Variations

As will be recognized by those skilled in the art, the innovative concepts described in the present application can be modified and varied over a tremendous range of applications, and accordingly the scope of patented subject matter is not limited by any of the specific exemplary teachings given, but is

10 only defined by the issued claims.

In particular, the disclosed inventions are not necessarily limited to "turbo decoding" as implemented in 1999. This is a very active area of technological development, and many improvements and modifications are undoubtedly in existence or soon to come. Even if future improved decoding

15 architectures with iterative processing of soft estimates are referred to by names other than "turbo decoding", the disclosed inventions may still be applicable.

For instance, multi-bit symbols could be used in lieu of binary symbols as the original data stream. Alternatively, future versions of iterative decoding might apply to analog signals rather than digital signals. The claimed

20 inventions can also be applied to decoder architectures that implement these changes.

Other decoding methods, such as pipeline decoding, or turbo codes that use more encoders than described in the present application may also be within the contemplation of the present innovations.

25 Further details and descriptions of the present state of the art may be found in TURBO CODING, by Heegard and Wicker, and ERROR CONTROL SYSTEMS, by Wicker, which are hereby incorporated by reference.